CLAIMS

What is claimed is:

1. A method of implementing a floating point compare operation in an x86 compatible processor, comprising:

comparing a first bit pattern and a second bit pattern using a floating point unit of an x86 compatible processor;

setting an EFLAGS register in accordance with a result of the comparison; and

encoding a sign flag and an overflow flag of the EFLAGS register with information derived from the result of the comparison.

2. The method of claim 1 wherein the first bit pattern and the second bit pattern are inputs for a floating point compare instruction.

3. The method of claim 1 wherein the floating point unit of the processor is configured to perform floating point operations in accordance with IEEE 754.

4. The method of claim 1 further comprising:

using combinational logic to encode the sign flag and the overflow flag based on the result of the comparison.

5. The method of claim 1 further comprising:

using the sign flag and the overflow flag encoded with information derived from the result of the comparison to implement a branch instruction, a conditional move instruction, a set operation, or a predicated operation.

6. A system for implementing a floating point compare operation in an x86 compatible processor, comprising:

a floating point unit configured to compare a first bit pattern and a second bit pattern;

an EFLAGS register, the EFLAGS register having a plurality of flags to receive a result of the comparison; and

encoding logic coupled to the EFLAGS register and the floating point unit, the encoding logic for encoding a sign flag and an overflow flag of the EFLAGS register with information derived from the result of the comparison.

7. The system of claim 6 wherein the first bit pattern and the second bit pattern are inputs for a floating point compare instruction.

8. The system of claim 6 wherein the floating point unit of the processor is configured to perform floating point operations in accordance with IEEE 754.

9. The system of claim 6 further comprising:

control logic for selectively enabling the encoding logic to encode the sign flag and the overflow flag with information derived from the result of the comparison.

10. The system of claim 6 wherein the sign flag and the overflow flag are encoded in accordance with the result of the comparison to implement a branch instruction, a conditional move instruction, a set operation, or a predicated operation.

11. A system for implementing an efficient floating point compare operation in an x86 compatible processor while ensuring x86 compatibility, comprising:

a floating point unit configured to compare a first bit pattern and a second bit pattern;

an EFLAGS register, the EFLAGS register having a plurality of flags to receive a result of the comparison;

encoding logic coupled to the EFLAGS register and the floating point unit, the encoding logic for encoding a sign flag and an overflow flag of the EFLAGS register with information derived from the result of the comparison; and

control logic for selectively disabling the encoding logic to encode the sign flag and the overflow flag with information derived from the result of the comparison to ensure x86 compatibility.

12. The system of claim 11 wherein the first bit pattern and the second bit pattern are inputs for a floating point compare instruction.

13. The system of claim 11 wherein the floating point unit of the processor is configured to perform floating point operations in accordance with IEEE 754.

14. The system of claim 11 wherein the control logic includes a multiplexer to selectively enable the encoding of the sign flag and the overflow flag in accordance with the result of the comparison.

15. The system of claim 11 wherein the sign flag and the overflow flag are encoded in accordance with the result of the comparison to implement a branch instruction, a conditional move instruction, a set operation, or a predicated operation.